

### Introduction

- ▶ Techniques have been developed to check multi-agent systems with a possibly unbounded number of agents.
- ▶ However, these techniques all assume the number of agents in the system is fixed in any given run.
- ▶ In many real-life systems, for example IOT applications and drone swarms, it is expected that agents will join and leave the system during its lifetime.

### Models

- ▶ We model our MAS by an agent template, which captures the possible states and behaviours of one of the agents leaving and joining, and an environment which captures the rest of the state of the system.
- ▶ An *agent* is a tuple  $A = \langle L, \iota, Act, P, t \rangle$  consisting of:
  - ▷ A set of local states  $L$ ,
  - ▷ A unique initial state  $\iota \in L$ ,
  - ▷ A non-empty set of actions  $Act$ ,
  - ▷ A protocol  $P : L \rightarrow \mathcal{P}(Act)$  that selects which actions may be performed at a given state,
  - ▷ A transition function  $t : L \times Act \times \mathcal{P}(Act) \times Act_E \rightarrow L$  that gives the agent's next state given its current state, the action performed by the agent, the set of actions performed by the other agents, and the action performed by the environment.
- ▶ We similarly define an environment.
- ▶ The OIS  $\mathcal{O} = \langle A, E, V \rangle$  is made up of an agent, an environment and a labelling function  $V : L \rightarrow \mathcal{P}(AP)$ . Transitions in it either respect the transitions of the agents and environment, or are special transitions in which an agent joins or leaves.
- ▶ We write  $\mathcal{O}(n)$  for the closed system of a fixed size  $n$  in which agents cannot join or leave.

### IAC<sub>TLK</sub>\X

- ▶ Given a set **IND** of indices, and a set **AP** of atomic propositions, the IAC<sub>TLK</sub>\X logic is the set of formulas  $\phi$  defined by the BNF:

$$\begin{aligned} \phi &::= \forall v: \phi \mid \psi \\ \psi &::= (p, v) \mid \neg(p, v) \mid \psi \wedge \psi \mid \psi \vee \psi \\ &\quad \mid A(\psi U \psi) \mid A(\psi R \psi) \mid K_v \psi \end{aligned}$$

where  $p \in \mathbf{AP} \cup \{a/v\}$ , and  $v \in \mathbf{IND}$ . Note we add a special atomic proposition  $a/v$  that holds whenever the agent is present in the system.

- ▶ We only consider formulas that are sentences, i.e., every variable appearing in the formula is in the scope of a universal quantifier.

### COIS

- ▶ A *collective open interpreted system* (COIS) is an OIS where the transition function of the agent satisfies

$$t(l, a, X \cup \{a\}, a_E) = t(l, a, X, a_E)$$

for all values of  $l, a, X$  and  $a_E$ .

- ▶ This means agents cannot transition differently based on whether their own action was part of the joint one, thus disallowing encoding of mutual exclusion.
- ▶ The problem of checking a COIS  $\mathcal{O}$  can be changed into the problem of checking the system  $\bar{\mathcal{O}}(n)$  for all  $n$ , where  $\bar{\mathcal{O}}$  is a transformed version of  $\mathcal{O}$ .
- ▶ This problem is decidable, so the model checking problem for COIS is decidable.

### IOIS

- ▶ IOIS are a subclass of OIS in which agents all actions are one of three types:
  - ▷ *Asynchronous*: Performed by one agent on its own.
  - ▷ *Agent-environment*: Performed by *one* agent together with the environment.
  - ▷ *Global synchronous*: Performed by *all* agents together with the environment.
- ▶ The problem of checking an IOIS  $\mathcal{O}$  can also be changed into the problem of checking a system  $\bar{\mathcal{O}}(n)$  for all  $n$ .
- ▶ However, the class that we transform IOIS into is only partially decidable, so the decision procedure is partial.
- ▶ In the paper, we identify a further subclass of IOIS for which we are guaranteed to have decidability.

### Implementation and Evaluation

- ▶ Our implementation, called MCMAS-OP, is based on MCMAS.
- ▶ We used our toolkit to model a train-gate controller example, in which the agents are trains wishing to enter a tunnel and the environment is a controller that has to ensure only one train is in the tunnel at any time. This was modelled as an IOIS.
- ▶ We also modelled a group of robots on a track that move forward at a fixed rate and have to stop at a certain point on the track but do not know their exact position. This was modelled as a COIS.
- ▶ We checked a few different properties for each system.

### Results

		Autonomous Robot (MCMAS-OP <sub>COIS</sub> )	Train-Gate Controller (MCMAS-OP <sub>IOIS</sub> )
Property	1	Satisfied	Satisfied
	2	Satisfied	Satisfied
	3	Satisfied	Not Satisfied
Model Build Time		30 sec	0.0023 sec
Memory Usage		199MB	31MB
Reachable States		1.49 million	45

Table: MCMAS-OP verification results obtained on our two examples, along with the build time, memory usage and number of reachable states for the model.

- ▶ The unsatisfied property is

$$\psi_3 \equiv \forall x: AG ((I_E = red) \rightarrow AF(I_E = green))$$

which says if the controller is in state red, then it will eventually be in state green (note a train leaving the system while in the tunnel will break this).

### Conclusion

- ▶ We have proposed a semantics to reason about open multi-agent systems in which agents can join and leave at run-time.
- ▶ While our verification problem is undecidable in general, we have presented two fragments that admit decision procedures and implemented these.
- ▶ We plan to continue work in this area by identifying further decidable fragments of the verification problem and building decision procedures for these.